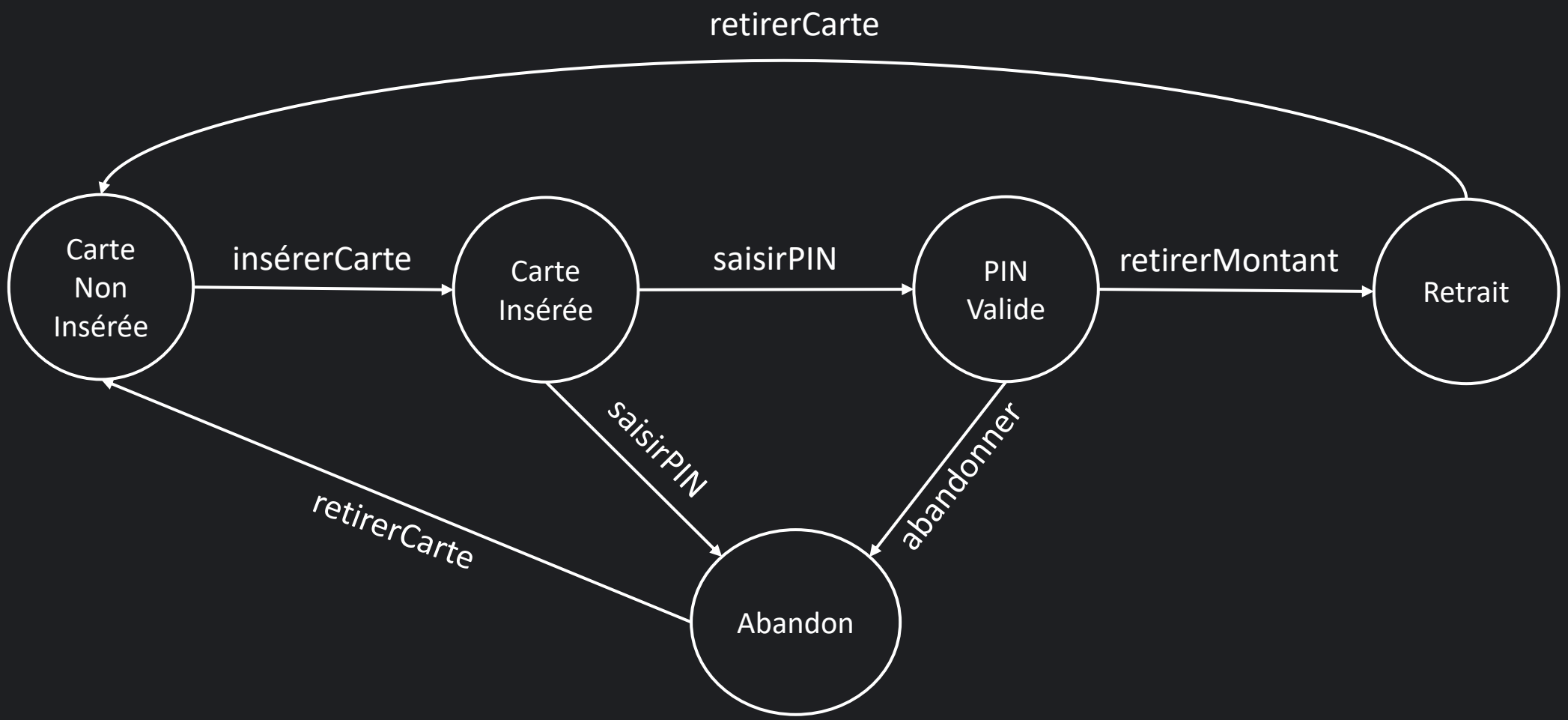
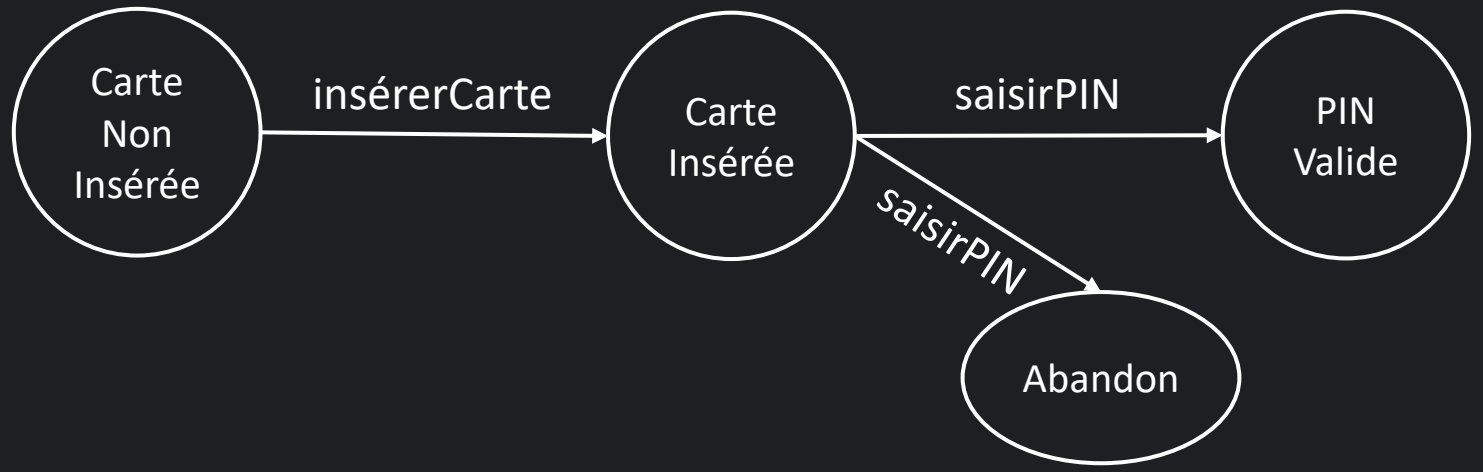
Patron Etat

Patrons de conception

Le problème

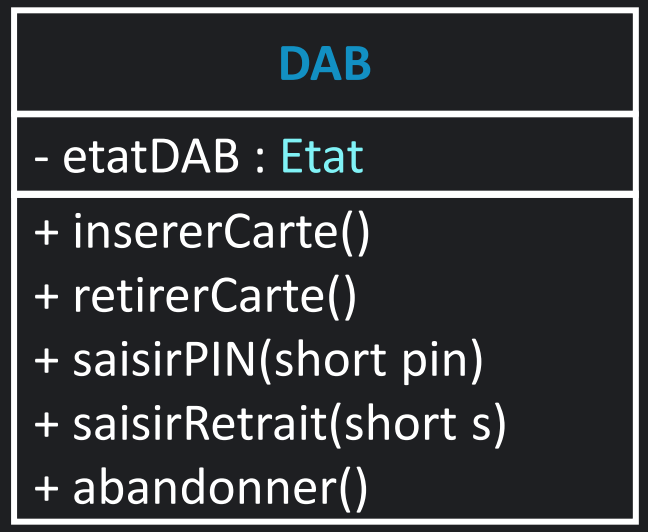

retirerCarte
Carte
Non
Insérée
insérerCarte
Carte
Insérée
saisirPIN
PIN
Valide
retirerMontant
Retrait
saisirPIN
abandonner
retirerCarte
Abandon


```
public void insererCarte() {
    switch (this.etat) {
    case ABANDON: /* Impossible */ break;
    case CARTE_INSEREE: /* Impossible */ break;
    case PIN_VALIDE: /* Impossible */ break;
    case RETRAIT: /* Impossible */ break;

    case CARTE_NON_INSEREE:
        etat = Etat.CARTE_INSEREE;
        break;

    default:
    }
}
```

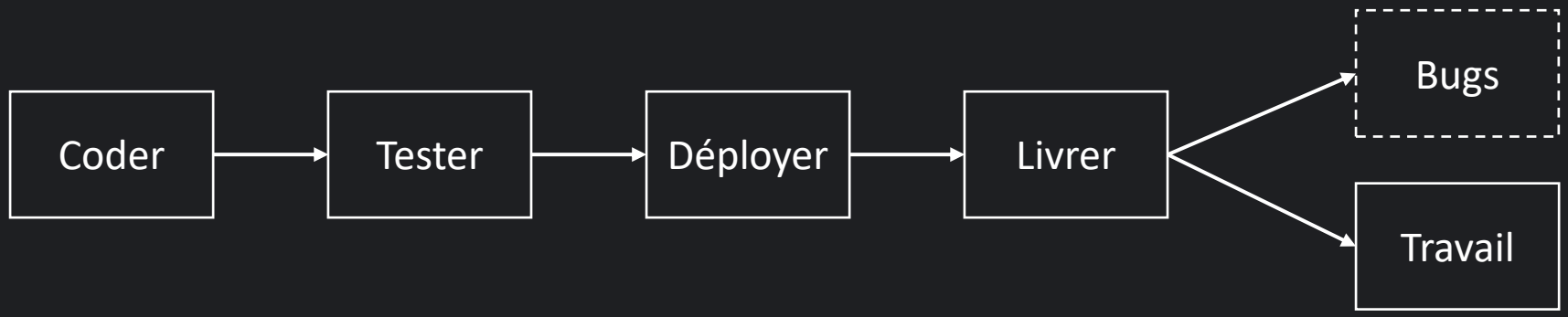



```java
public void insererCarte() {
    switch (this.etat) {
    case ABANDON: /* Impossible */ break;
    case CARTE_INSEREE: /* Impossible */ break;
    case PIN_VALIDE: /* Impossible */ break;
    case RETRAIT: /* Impossible */ break;

    case CARTE_NON_INSEREE:
        etat = Etat.CARTE_INSEREE;
        break;

    default:
    }
}
```

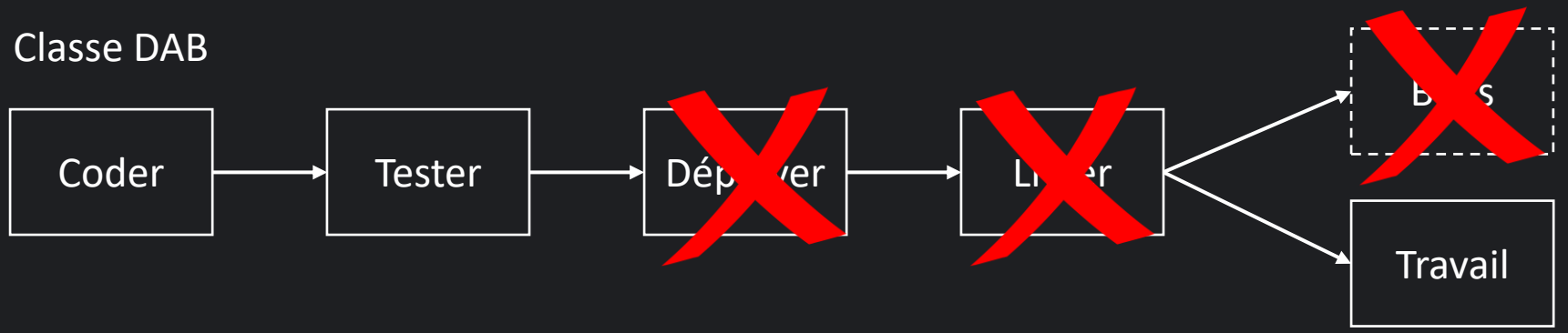
```java
public void saisirPin(short pin) {
    switch (this.etat) {
    case ABANDON: break;
    case CARTE_NON_INSEREE: break;
    case PIN_VALIDE: break;
    case RETRAIT: break;
    case CARTE_INSEREE:
        if(pin == 1234) {
            etat = Etat.PIN_VALIDE;
        } else {
            etat = Etat.ABANDON;
        }
        break;
    default:
    }
}
```



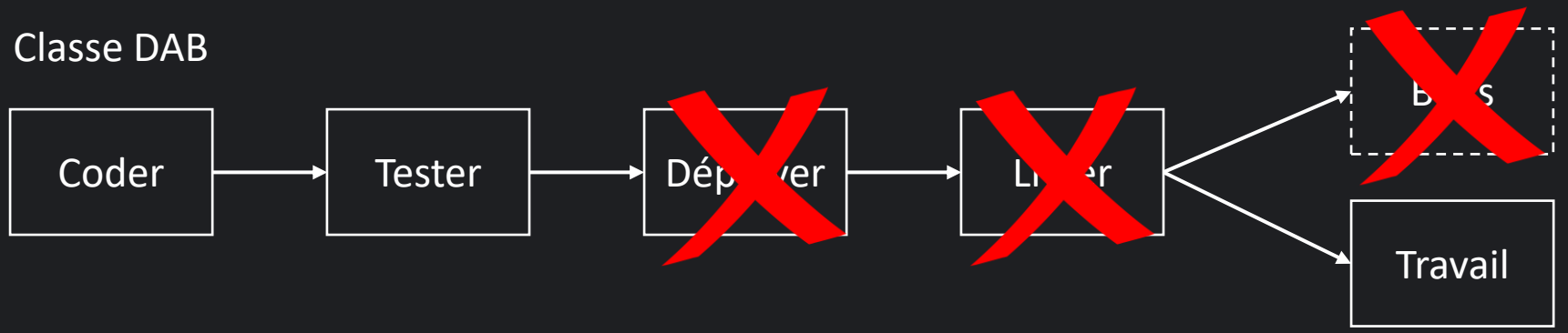

```java
public void saisirPin(short pin) {
    switch (this.etat) {
    case ABANDON: break;
    case CARTE_NON_INSEREE: break;
    case PIN_VALIDE: break;
    case RETRAIT: break;
    case CARTE_INSEREE: ... break;

    case NOUVEAU_ETAT:

    default:
    }
}
```



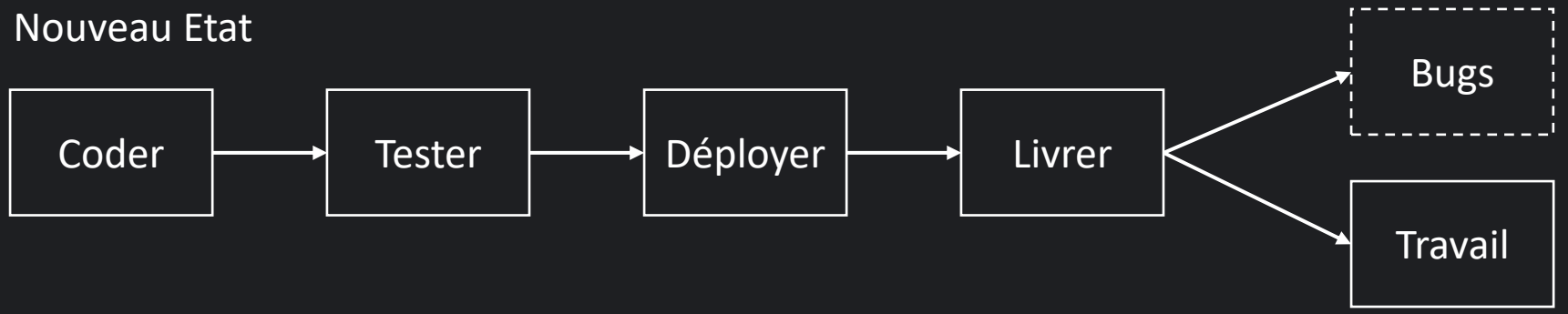
Classe DAB



Classe DAB



Nouveau Etat



Je veux

Ouvert à l'extension et fermé à la modification

Principe Ouvert/Fermé

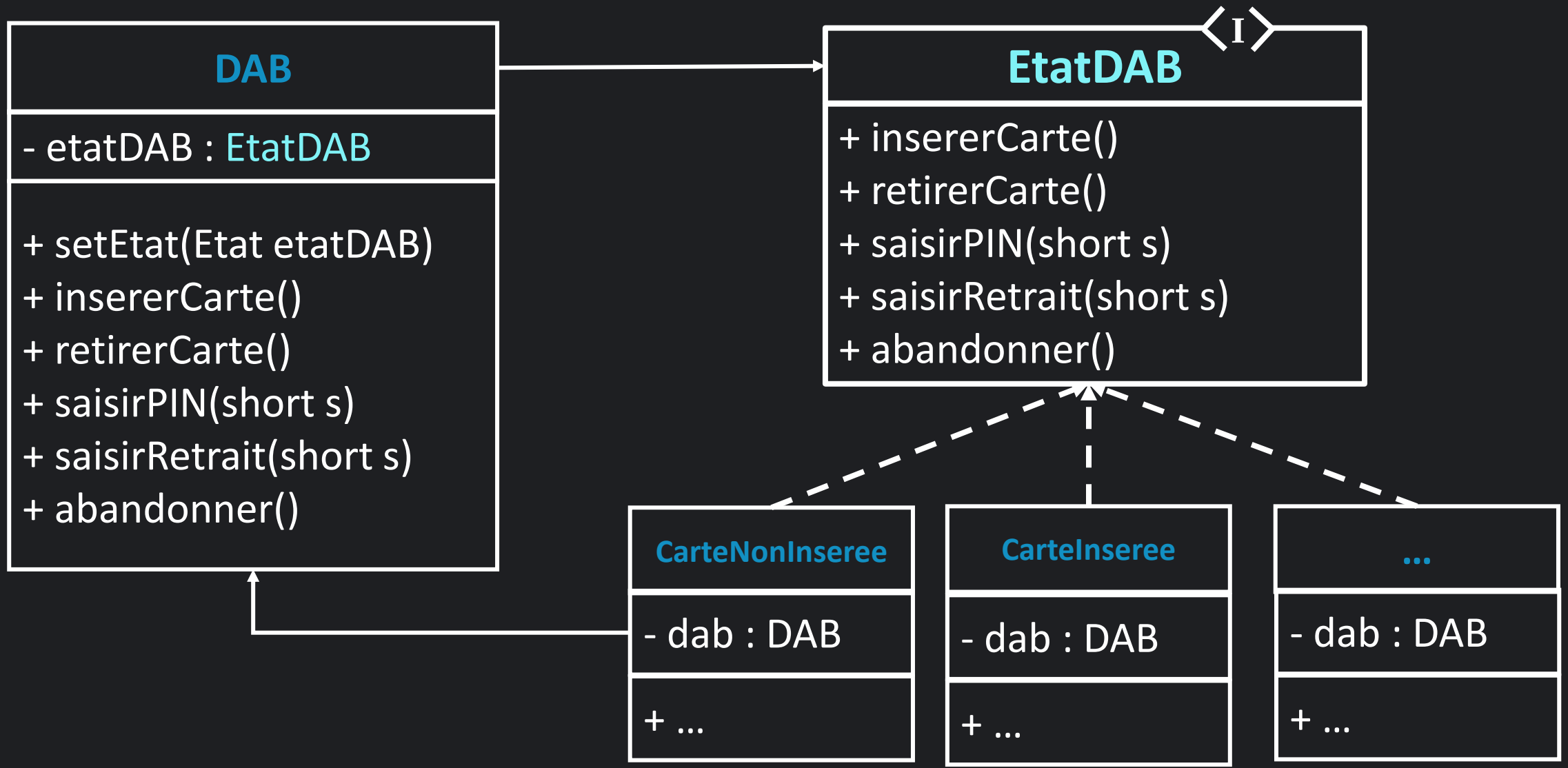
Patron état

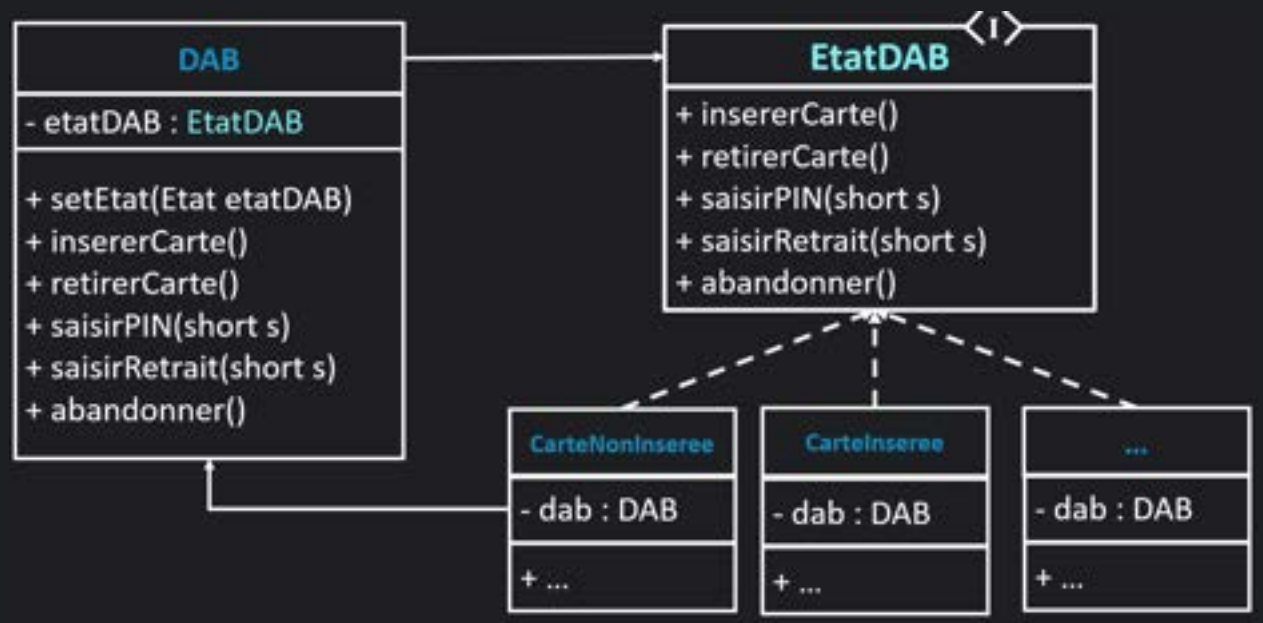
Définition et mise en place

> Pouvoir changer le comportement d'un objet quand son état change, sans pour autant en changer l'instance
>
> Patron état

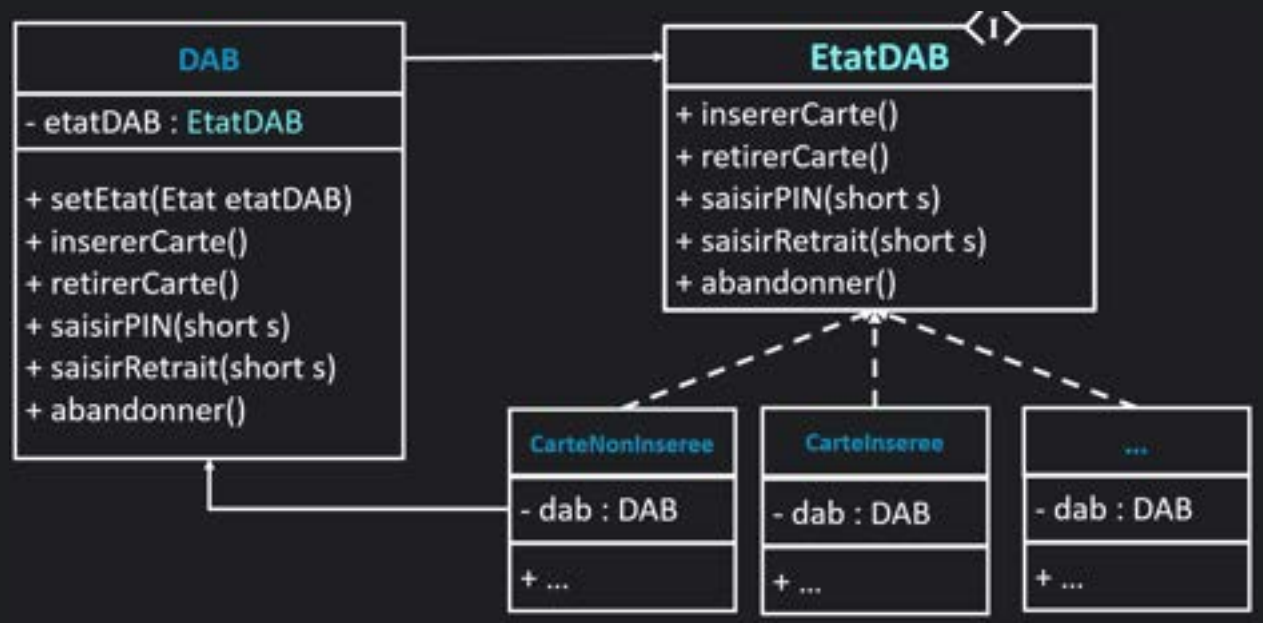
Pouvoir changer le comportement du **DAB** quand son état change, sans pour autant en **changer d'instance DAB**

Dans notre cas


DAB
- etatDAB : EtatDAB
+ setEtat(Etat etatDAB)
+ insererCarte()
+ retirerCarte()
+ saisirPIN(short s)
+ saisirRetrait(short s)
+ abandonner()
I
EtatDAB
+ insererCarte()
+ retirerCarte()
+ saisirPIN(short s)
+ saisirRetrait(short s)
+ abandonner()
CarteNonInseree
- dab : DAB
+ ...
CarteInseree
- dab : DAB
+ ...
...
- dab : DAB
+ ...

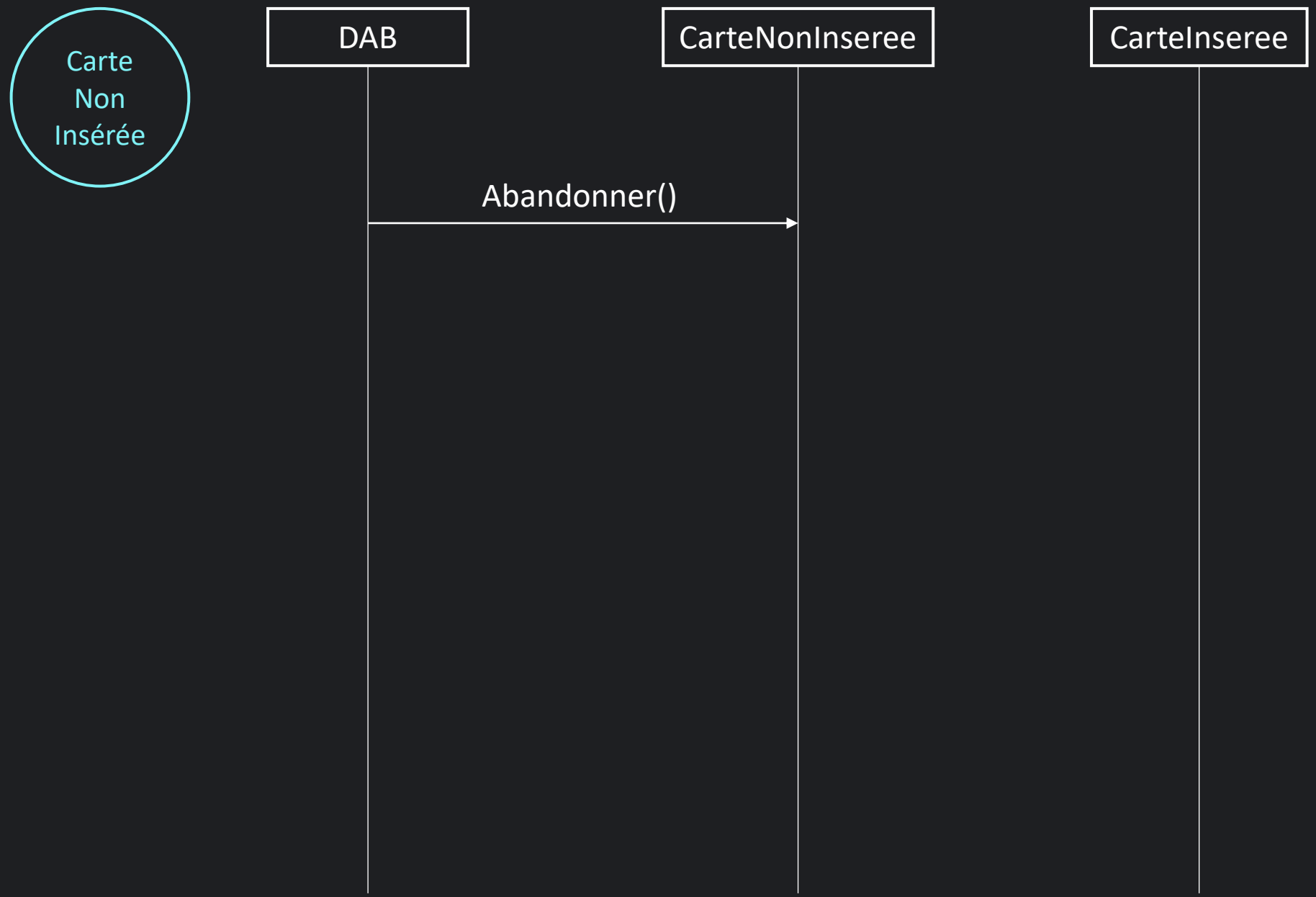


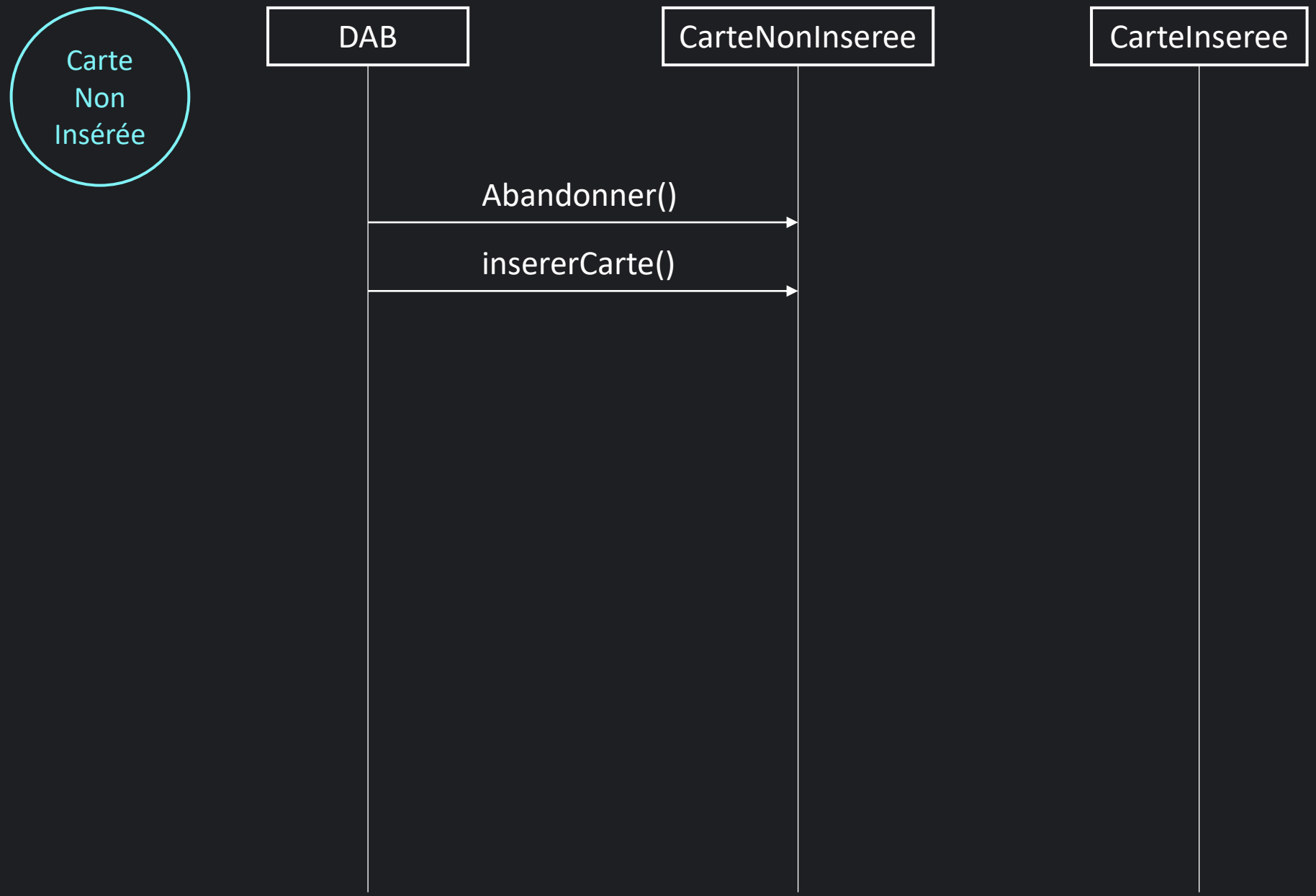

```
public class DAB {
  private Etat etat;

  public DAB() {
      etat = new CarteNonInseree(this);
  }

  void abandonner() { etat.abandonner(); }
  void saisirCode(short code) { etat.saisirCode(code); }
  ...
}
```

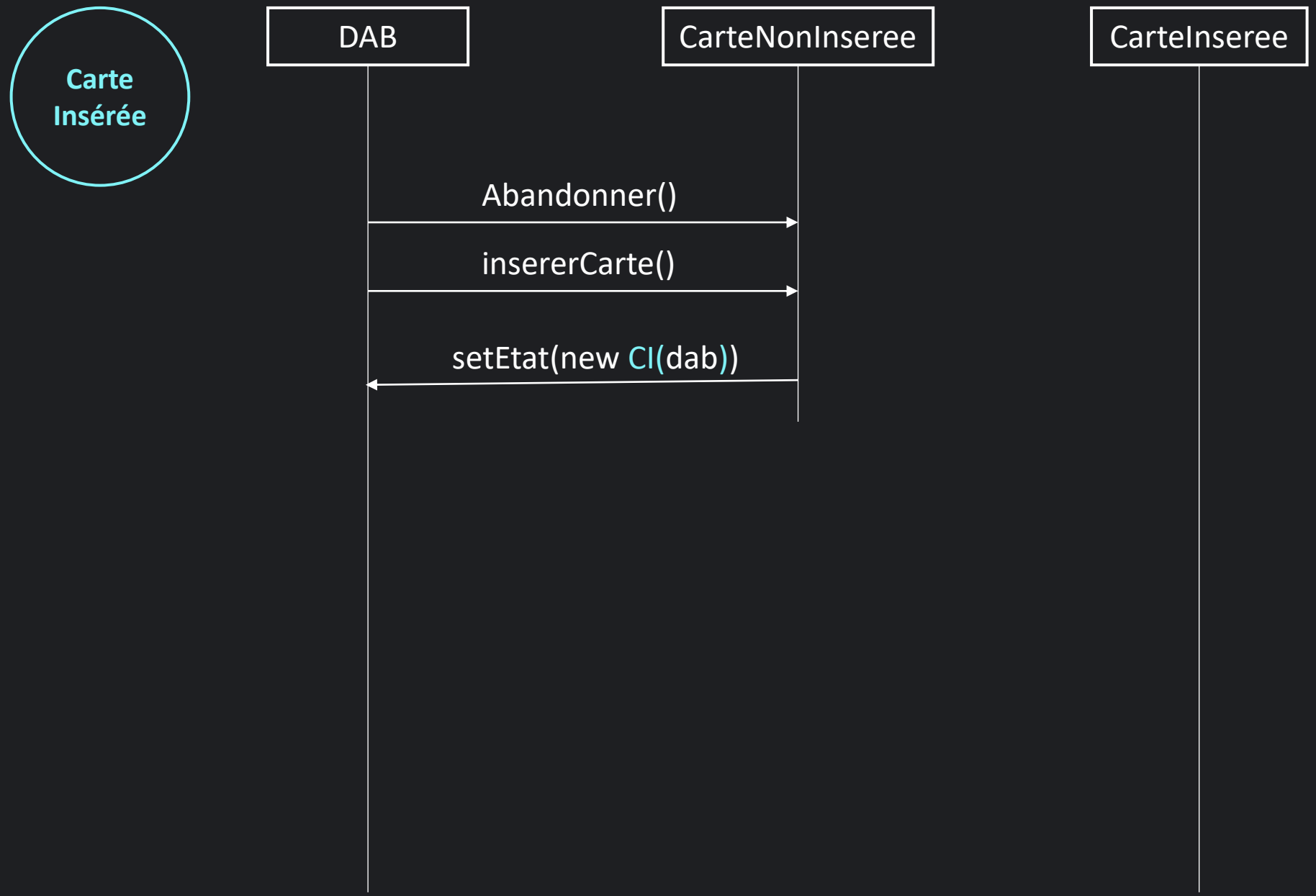


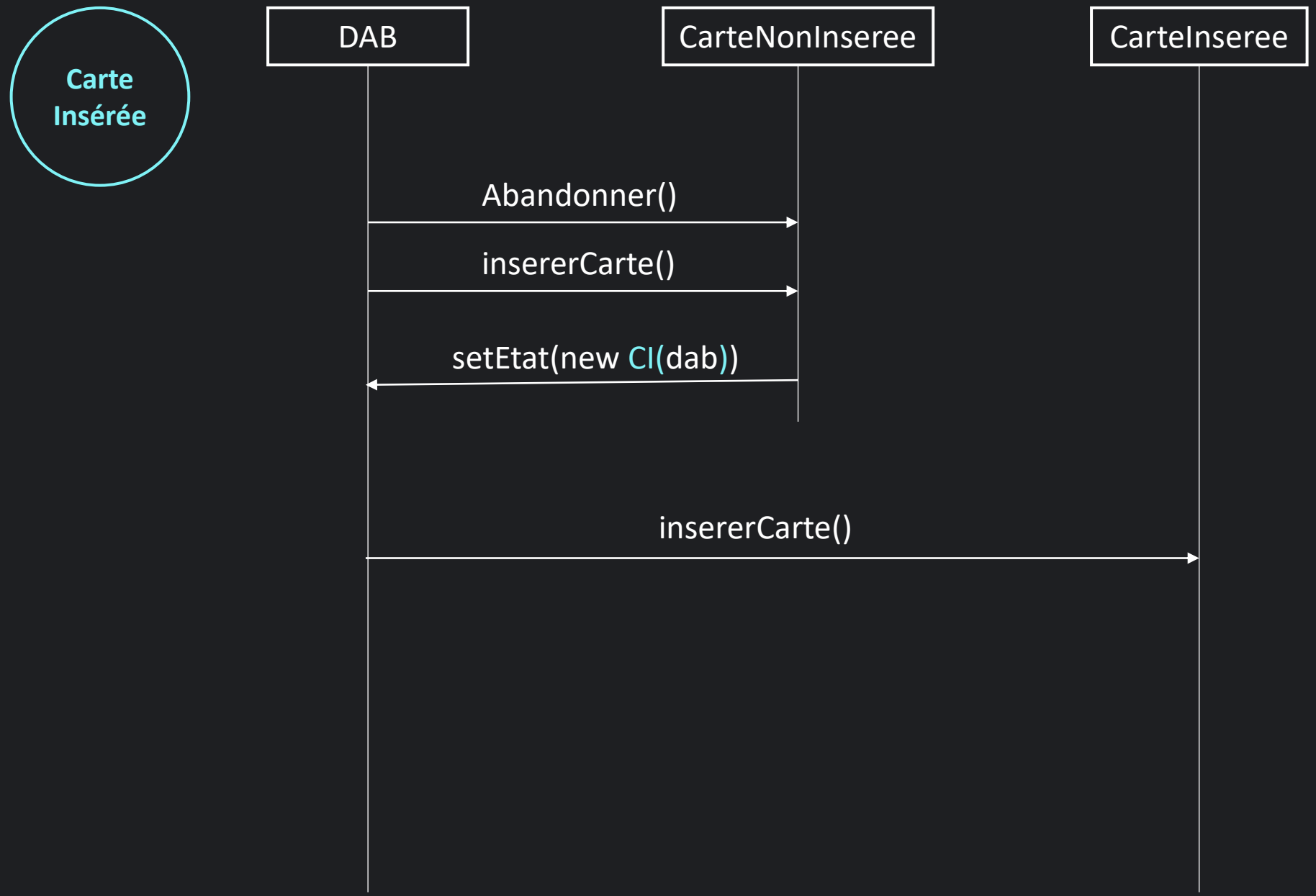

```java
public class DAB {
  private Etat etat;

  public DAB() {
      etat = new CarteNonInseree(this);
  }

  void abandonner() { etat.abandonner(); }
  void saisirCode(short code) { etat.saisirCode(code);
  ...
}
```

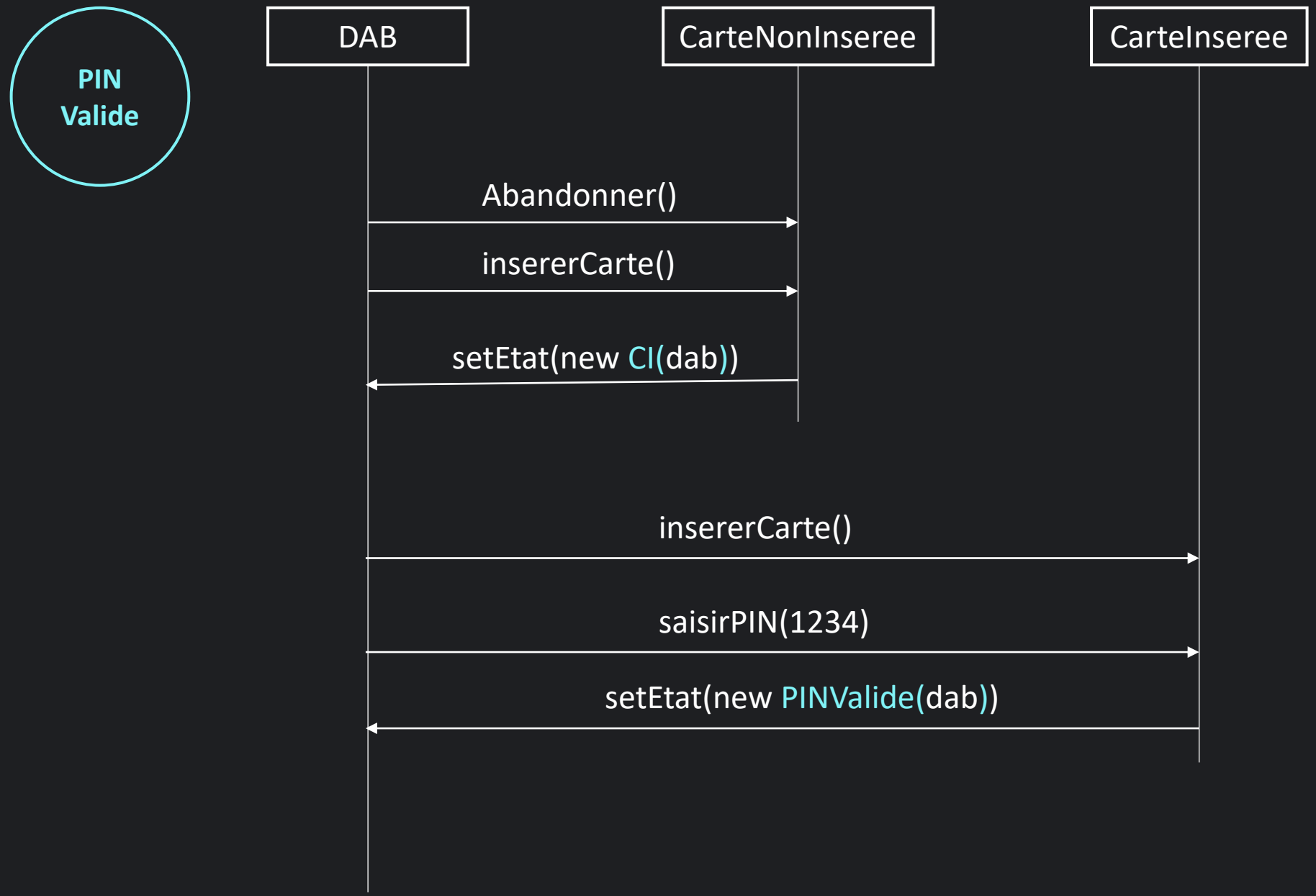
```java
public class CarteNonInseree implements Etat {

  private DAB dab;

  public CarteNonInseree(DAB dab) {
    this.dab = dab;
  }

  void abandonner() { /* Impossible */ }
  void insererCarte() { dab.setEtat(new CarteInseree(dab)); }
}
```


Carte
Non
Insérée
DAB
CarteNonInseree
CarteInseree
Abandonner()


Carte
Non
Insérée
DAB
CarteNonInseree
CarteInseree
Abandonner()
insererCarte()


Carte Insérée
DAB
CarteNonInseree
CarteInseree
Abandonner()
insererCarte()
setEtat(new CI(dab))


Carte Insérée
DAB
CarteNonInseree
CarteInseree
Abandonner()
insererCarte()
setEtat(new CI(dab))
insererCarte()


PIN Valide
DAB
CarteNonInseree
CarteInseree
Abandonner()
insererCarte()
setEtat(new CI(dab))
insererCarte()
saisirPIN(1234)
setEtat(new PINValide(dab))

Mise en œuvre

Sans patron Etat

Mise en œuvre

Avec patron Etat

Désavantage

- Obliger d’implémenter toutes les méthodes de l’interface

```
public class Abandon implements Etat {

  private DAB dab;

  public void abandonner() { /* IMPOSSIBLE */ }

  public void insererCarte() { /* IMPOSSIBLE */ }

  public void saisirCode(short code) { /* IMPOSSIBLE */ }

  public void saisirMontant(short code) { /* IMPOSSIBLE */ }

  public void retirerCarte() {
    dab.setEtat(new CarteNonInseree(dab));
  }
}
```

- Obliger d'implémenter toutes les méthodes de l'interface

```
public class Abandon extends EtatAbs1 {

  private DAB dab;

  public void saisirCode(short code) { /* IMPOSSIBLE */ }

  public void saisirMontant(short code) { /* IMPOSSIBLE */ }

  public void retirerCarte() {
    dab.setEtat(new CarteNonInseree(dab));
  }
}
```